Exercise :Git Installation - Basic Commands

Step 1: Installation of Git on windows and EC2 Instance

Step 1.1: Git On EC2 Instance

Create a new EC2 instance (2018-03) with ssh as security group inbound rule and login to the instance with SmartTTY tool and run the following commands

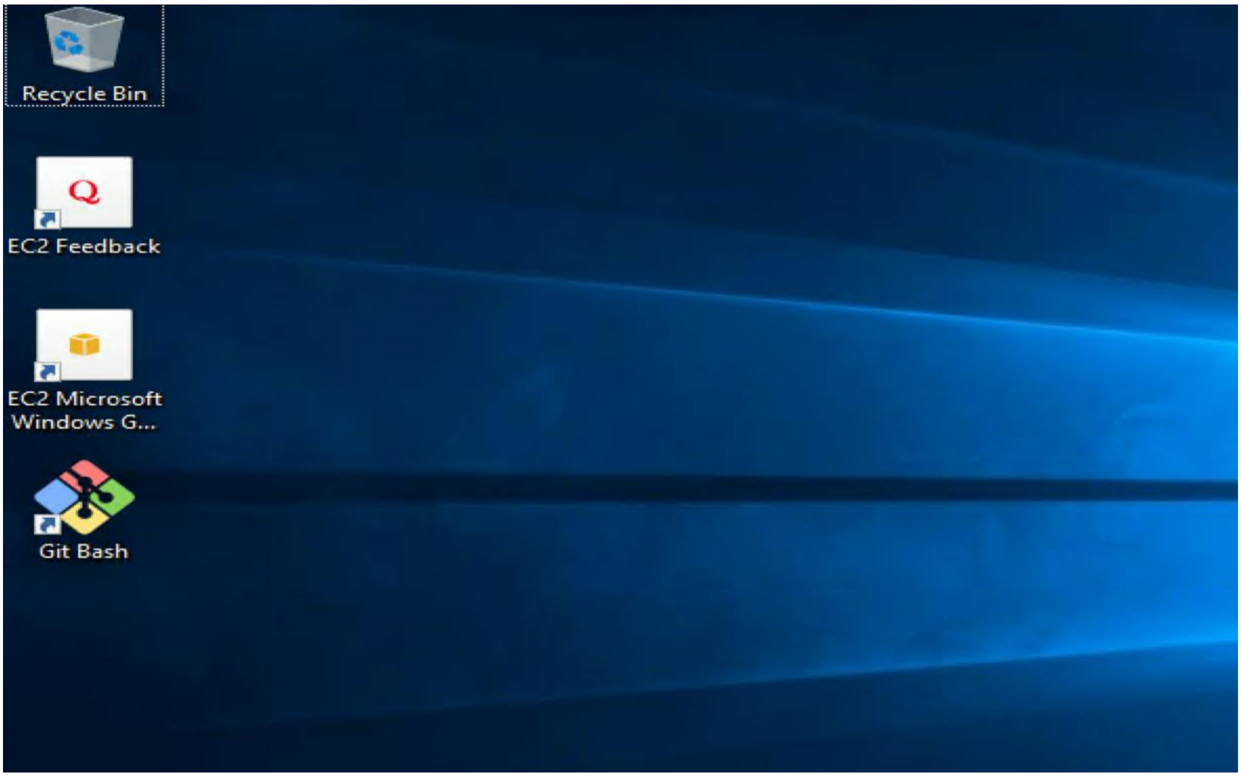
```
$ sudo yum install git -y
$ git version
```

Step 1.2: Git on Windows

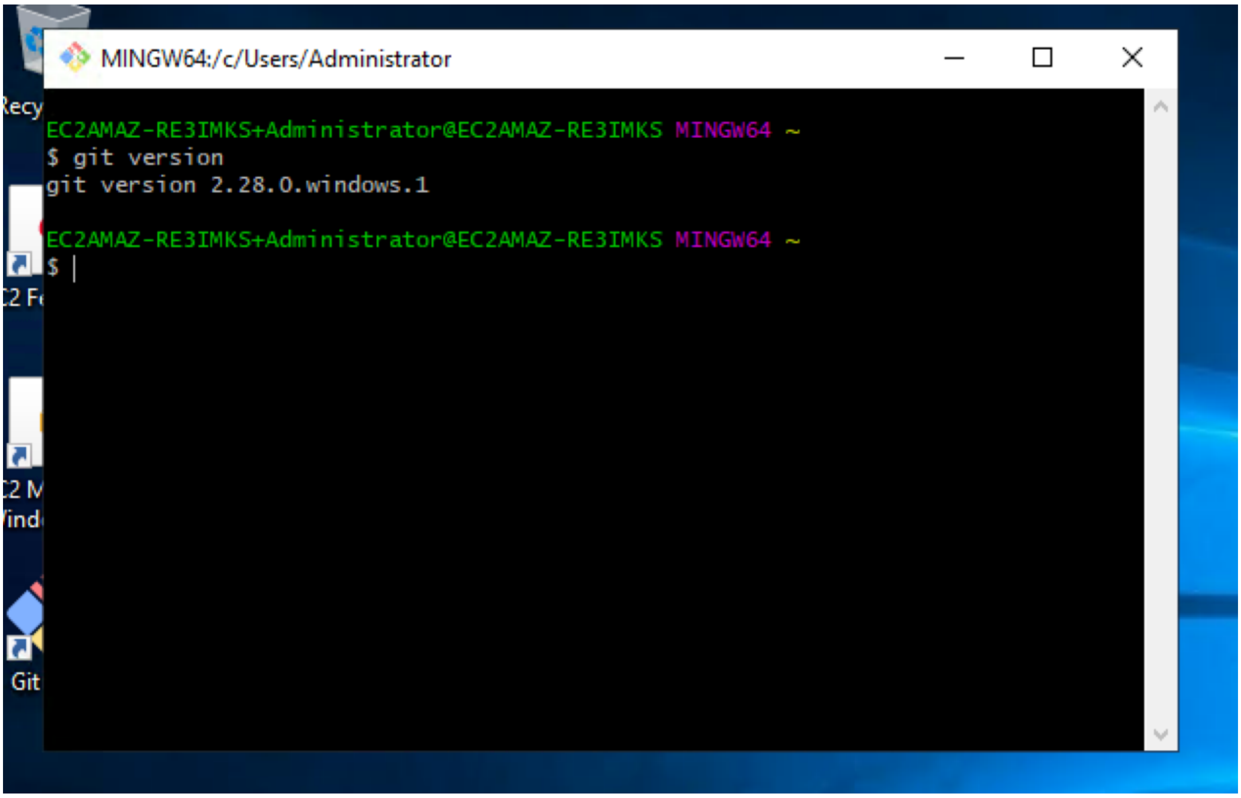
Download and install git from the following url and go with all default options

https://gitforwindows.org/

Double click the Git bash icon on the desktop



You should see a command window where you can type Git commands. Type Git version to check the version of Git installed



```
EC2AMAZ-RE3IMKS+Administrator@EC2AMAZ-RE3IMKS MINGW64 ~
$ git version
git version 2.28.0.windows.1

EC2AMAZ-RE3IMKS+Administrator@EC2AMAZ-RE3IMKS MINGW64 ~
$ |
```

Step 1.3: Configure Git using the following commands

```
$ git config --global user.name "Your name"
$ git config --global user.email “your email"
$ git config --list
```

Step 2: Initializing a git repository

```
$ mkdir myproject
$cd myproject
$ git init
```

Step 3: Adding files to git and commit

```
$cd myproject
$ git status
$ echo "My first file in git" >file1.txt
$ git status
$ git add .                (to add all files)
$git status
$ git commit -m "my first commit”
```

Step 4: Modify files in git and commit

```
$echo "My first file changed" >>file1.txt
$git status
$git add .
$git commit –m "new changes"
```

Step 5: Viewing commit logs

```
$echo "My second file " >file2.txt
$echo "My third file " >>file3.txt
$ git log
```

Step 6: To view commit logs of a specific user

```
$ git log --author "karthik"
```

Step 7: Add the new files to git

```
$ git add .
$ git status
$ git commit -m "New files added"
$ git log
```

Step 8: To back out any changes made to existing files

```
$ echo “Initial text”> sample.txt
$ git add .
$ git commit -m “New commit”
$ echo “More text”>> sample.txt
$ git status
$ git add .
//To revert the changes from staging area
$ git reset HEAD sample.txt
//To revert the changes from unstaged area
$ git checkout  -- sample.txt
$ git status
//To revert the changes from local repo
$ git log
$ git revert <Commit id”>
```

Step 9: Viewing the commit logs

```
$ git log
$ git log --oneline --graph –decorate
$ git log --since=”2 days ago”
```

Step 10: To show all the commits of a file

```
$ git log <file name>
```

Step 11: To show the details of a commit

```
$ git show <commit id>
```

Step 12: To ignore files from project for commits

Create a file with name .gitignore

Add the following patterns to file to ignore

```
Testfile.txt   // to just ignore a file
*.log           // to ignore a pattern
testdir/         // to ignore a directory
```

Step 13: To show the differences in working directory

```
$ git diff [file name]
```

Step 14: To show the differences between working directory and Local repo

```
$ git diff HEAD
```

Step 15: To show the differences between staging directory and Local repo

```
$ git diff --staged HEAD
```

Step 16: To compare 2 commits of a file

```
$ git log –oneline
$ git diff <commitid 1> <commitid 2>
```

Step 17: To compare last 2 commits of a local repo

```
$ git diff HEAD HEAD^
```